# **[设计模式--对象池模式](https://segmentfault.com/a/1190000022958322)**

IMG_256

今天我们来介绍一下对象池模式，这种设计模式在平时的工作中使用频率不像工厂模式那样使用频繁，但是它提供了一种新的对象使用和管理思想，还是很值得借鉴的。

既然提到池了，那就是某一类或者一种事物的集合，比如线程池，这里面会涉及到把对象放到池子和从池子里面取对象的过程。

那我们先来定义一个Art类：

**class** **Art**{

**private** $\_name;

*/\*\**

*\* @return mixed*

*\*/*

**public** **function** **getName**()

{

**return** **$this**->\_name;

}

*/\*\**

*\* @param mixed $name*

*\*/*

**public** **function** **setName**($name)

{

**$this**->\_name = $name;

}

}

我们还需要定义一个对象池（ArtPool）,来处理Art的入池和出池操作：

**class** **ArtPool**{

*//艺术类列表*

**static** **private** $\_arts = [];

*/\*\**

*\* 对象池中加入对象*

*\**

*\* @param Art $art*

*\*/*

**static** **public** **function** **pushArt**(Art $art)

{

**if**(!**isset**(**static**::$\_arts[$art->getName()]) || **empty**(**static**::$\_arts[$art->getName()]))

{

**static**::$\_arts[$art->getName()] = $art;

}

}

*/\*\**

*\* 获取对象*

*\**

*\* @param $name*

*\* @return mixed|null*

*\*/*

**public** **static** **function** **getArt**($name)

{

**return** **isset**(**self**::$\_arts[$name]) ? **self**::$\_arts[$name] : **null**;

}

*/\*\**

*\* 从对象池中删除对象*

*\**

*\* @param $name*

*\*/*

**public** **static** **function** **removeArt**($name)

{

**if**(array\_key\_exists($name, **self**::$\_arts))

{

**unset**(**self**::$\_arts[$name]);

}

}

}

这里面有个比较重要的地方需要注意一下，就是:

*//艺术类列表***static** **private** $\_arts = [];

这个静态变量来存储所有的Art对象；

调用过程和执行结果：

**class** **Client**{

**public** **static** **function** **main**()

{

$music = **new** Art();

$music->setName('music');

*//加入到对象池中*

ArtPool::pushArt($music);

$movie = **new** Art();

$movie->setName('movie');

ArtPool::pushArt($movie);

$music\_in\_pool = ArtPool::getArt('music');

p('$music\_in\_pool',$music\_in\_pool);

$movie\_in\_pool = ArtPool::getArt('movie');

p('$movie\_in\_pool',$movie\_in\_pool);

}

}

执行结果：

\*\*\*\*\*\*\*

$music\_in\_pool=**object**(Art)**#2 (1) {**

[**"\_name":"Art":private**]=>

**string**(5) "music"

}

\*\*\*\*\*\*\*

$movie\_in\_pool=**object**(Art)**#3 (1) {**

[**"\_name":"Art":private**]=>

**string**(5) "movie"

}

我们根据不同的name来获取不用的Art对象，这样节省了new 时候的内存和操作时间，主要适用于频繁创建对象和销毁对象的场景。

就像开始说，对象池模式相对比较简单，但是给我们提供了创建对象时一种比较有意思的思路。我们也可以在创建池对象的时候，就创建整个对象集合，这样还可以减少内存碎片，提高内存的使用效率。

phper 在进阶的时候总会遇到一些问题和瓶颈，业务代码写多了没有方向感，不知道该从那 里入手去提升，对此我整理了一些资料，包括但不限于：分布式架构、高可扩展、高性能、 高并发、服务器性能调优、TP，laravel，Redis，Swoole、Swoft、Kafka、Mysql 优化、Docker、微服务、Nginx 等多个知识点高级进阶干货需要的可以免费分享给大 家 需要可以加下我 qq:1047403189 或者微信：PHPty123获取哦

这些的话也在下面的直播课中讲解过很多期，并且在不断更新中，是结合企业的一些应用场景讲解的，能够帮助学员突破思维或者带着实战，来帮助大家掌握学习的电脑 腾讯课堂客户端

手机 腾讯课堂APP

网页版直播地址：https://ke.qq.com/course/328509?tuin=6

2020年 6月23日 晚上20-23点 主讲老师介绍:

# 六星教育-Peter老师

六星PHP学院院长、六星教育合伙人，9年PHP研发经验，曾就职于拓维、新浪等上市公司，经历过大型项目的攻坚与调优，擅长领域：接口服务设计，微服务架构、swoole网络编程、机器学习，熟悉Laravel等框架的优化；对大数据业务、高并发数据优化拥有丰富的经验。